IR simulations update: luminosity monitor (mostly)

Richard Petti (BNL)

EIC Task Force Meeting

05-07-2015

Outline

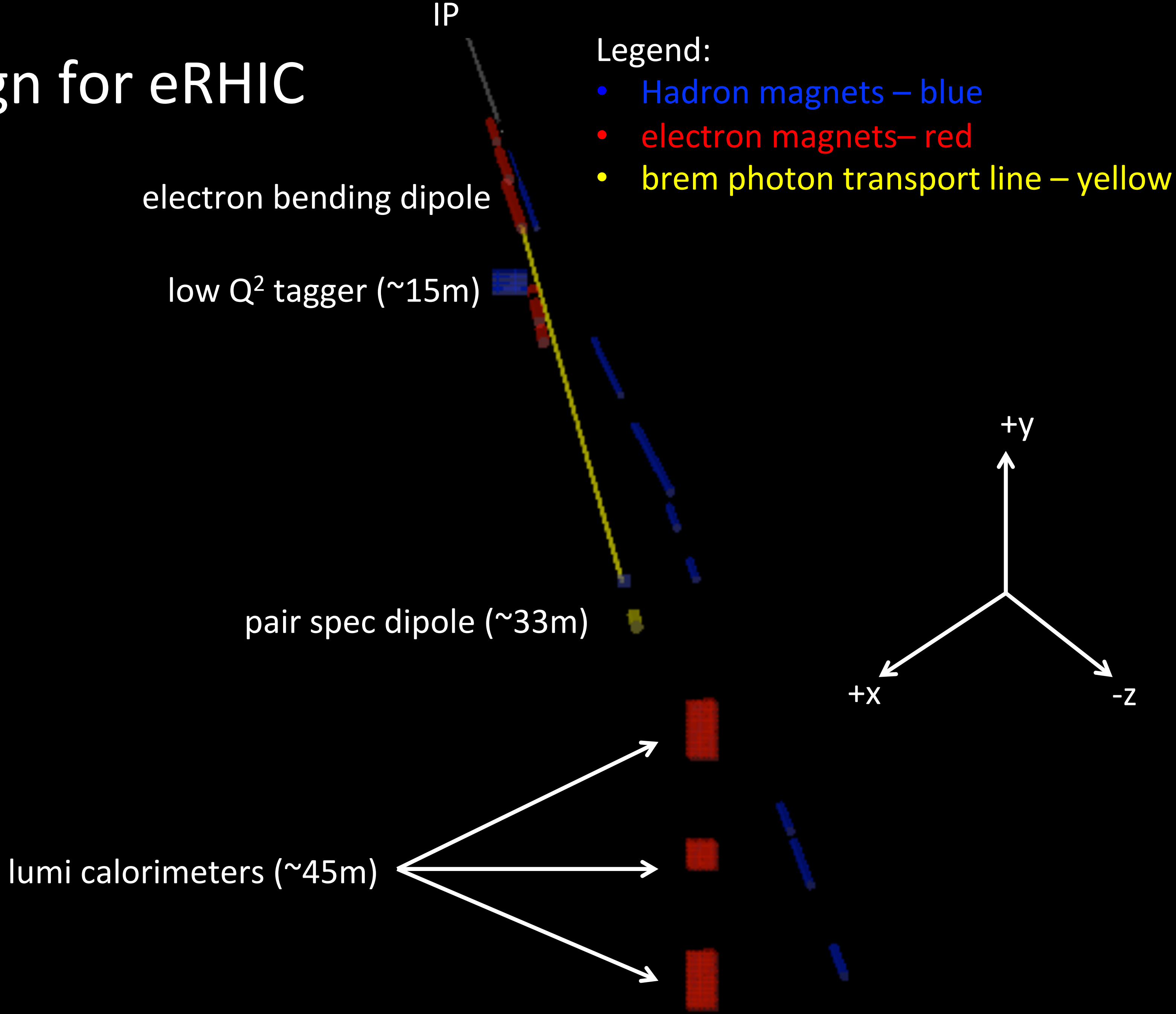
- generalities of a luminosity monitor
- reminder of luminometer design at eRHIC
- quantification of the required acceptance
- notable code developments
  - DJANGOH
  - eicsmear
  - EicRoot

Measuring Luminosity

- closely follow the implementation by ZEUS (HERA II)
- measure the process e+p -> e+p+gamma
  - high rate
  - pure QED with precisely calculable cross section
  - luminosity related to measured rate of photons

$$L = \frac{N_\gamma}{A\sigma}$$

A design for eRHIC

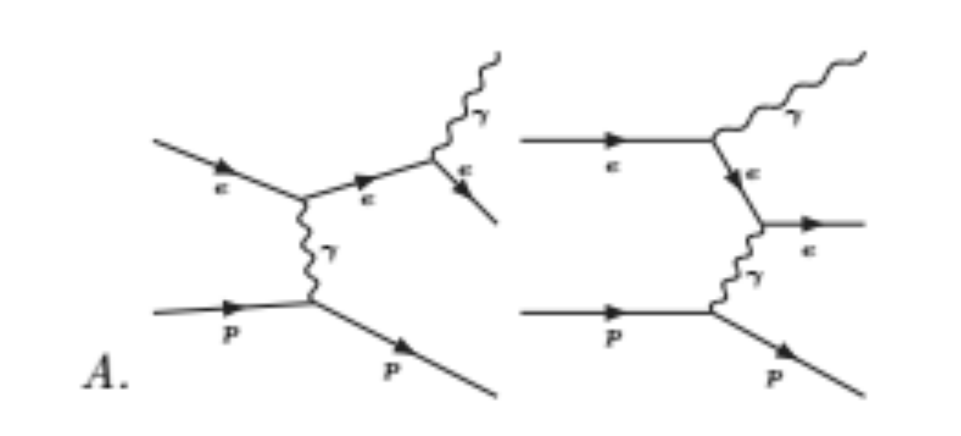


Requirements for the acceptance of photons

- need to give CAD a number for the angular acceptance needed for photons
- requirement coming from:
  - the angular distribution of photons coming directly from the physics process
  - the angular spread of said distribution from beam effects (emittance, crossing angle, etc.)

Distribution of photons from physics

- Bethe-Heitler calculation (1934)



energy distribution

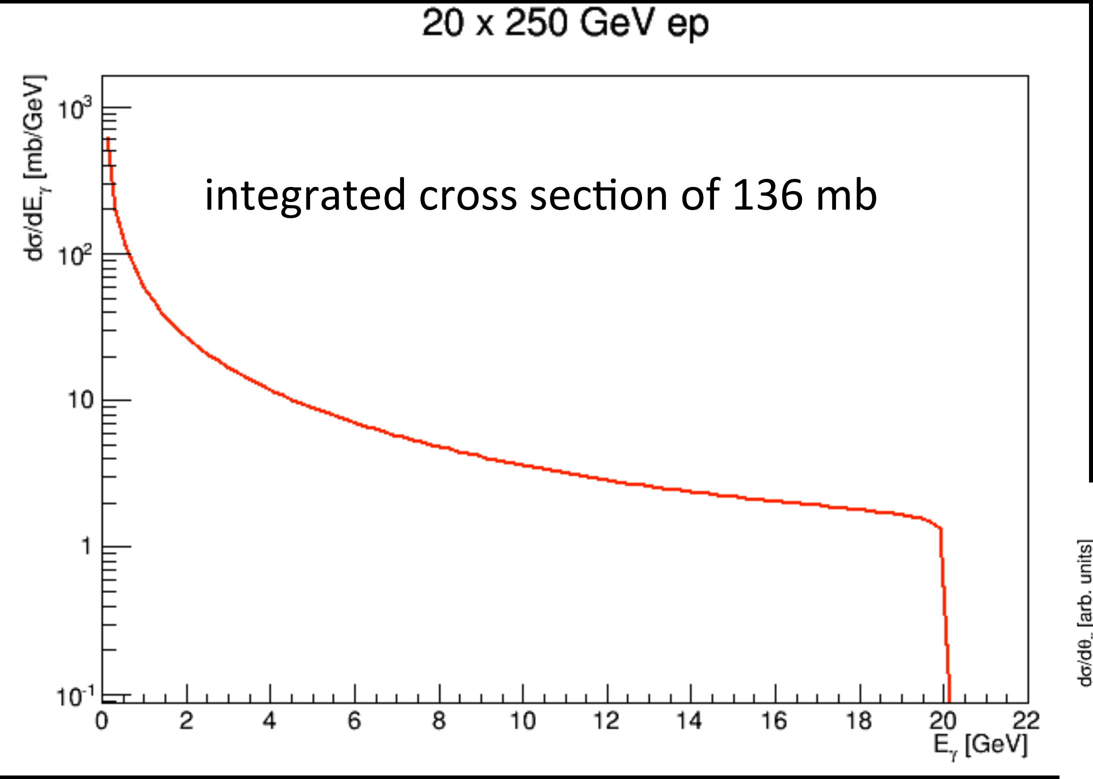
$$\frac{d\sigma}{dE_\gamma} = 4\alpha r_e^2 \frac{E_e'}{E_\gamma E_e}\left(\frac{E_e}{E_e'} + \frac{E_e'}{E_e} - \frac{2}{3}\right)\left(\ln\frac{4E_p E_e E_e'}{m_p m_e E_\gamma} - \frac{1}{2}\right)$$

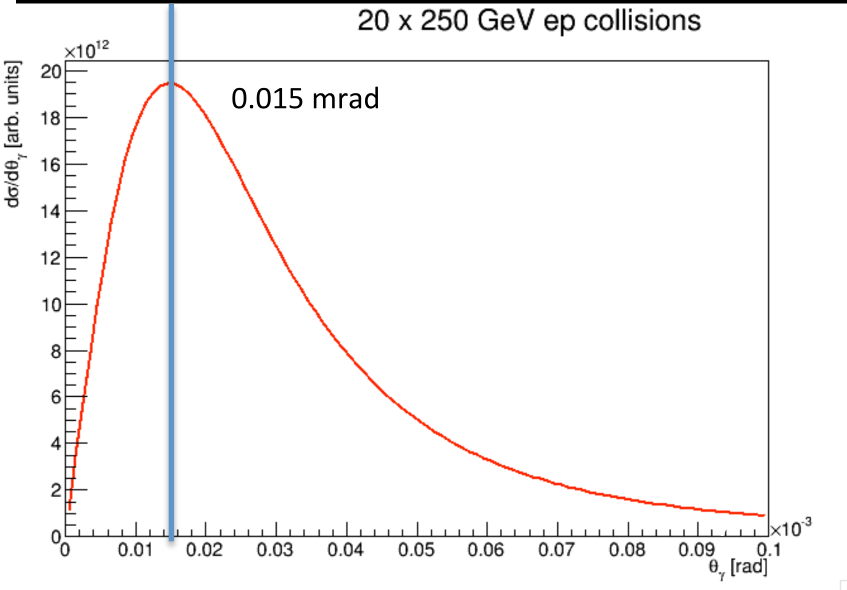
angular distribution

$$\frac{d\sigma}{d\Theta_\gamma} \approx \frac{\Theta_\gamma}{\left(\left(m_e/E_e\right)^2 + \Theta_\gamma^2\right)^2}$$

How do the distributions look for eRHIC collisions?

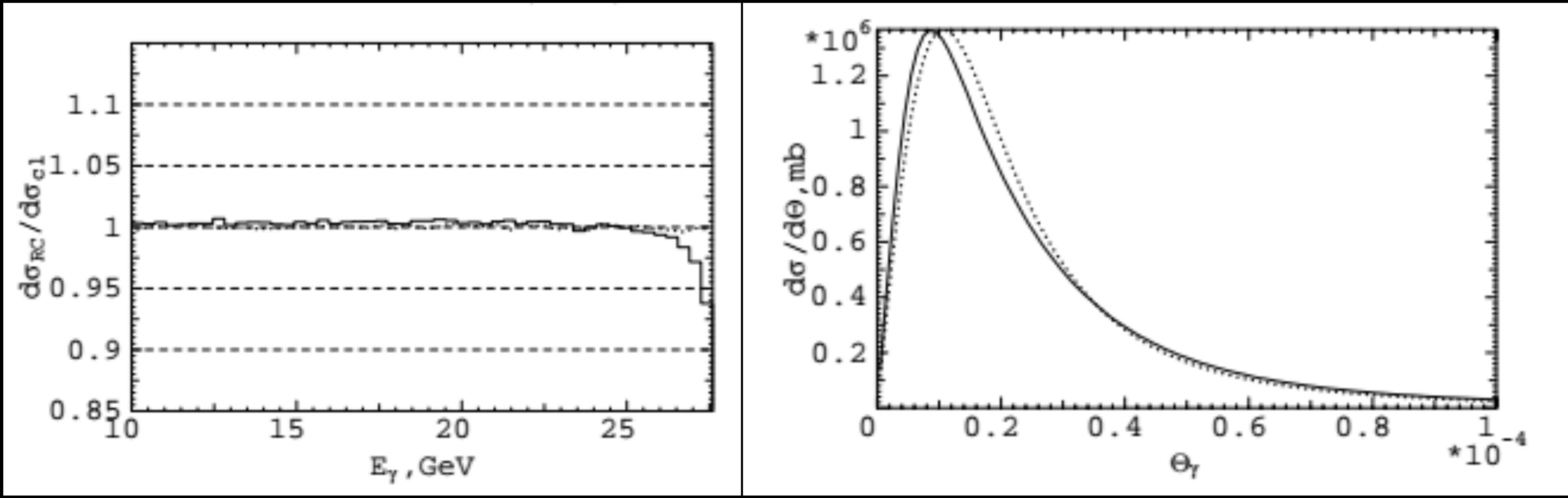
- code obtained from B. Schmidke to generate these distributions





Effect of radiative corrections

- [1] Precision calculation of processes used for luminosity measurement at the ZEUS experiment, T.Haas, V.Makarenko, arXiv:1009.2451v2
- include various one-loop QED radiative corrections
- note: the following is a comparison of calculations done at 27.6 x 920 GeV ep collisions
- corrections are small



DJANGOH simulations

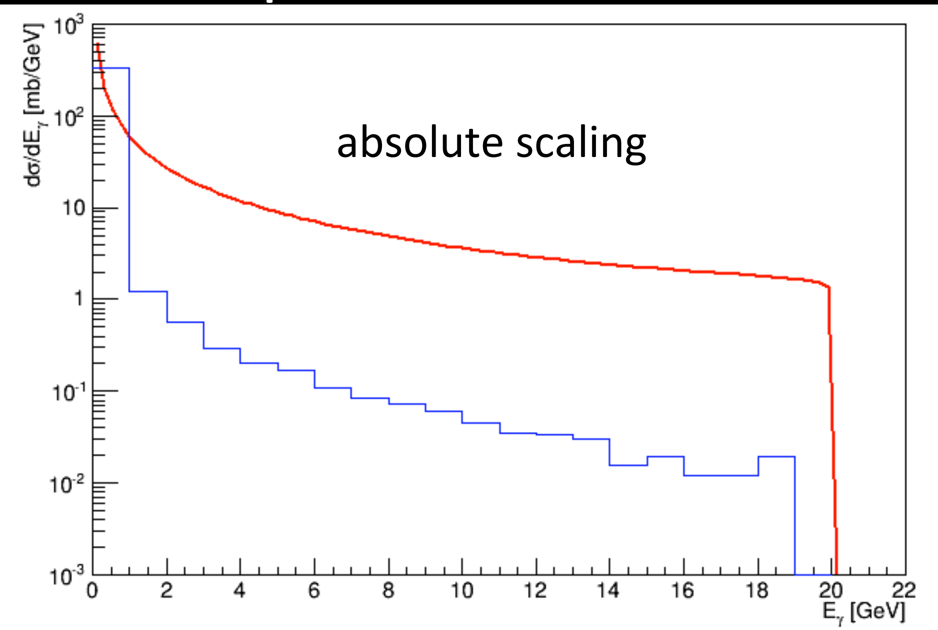
- use DJANGOH to study effect of rates in polarized ep collisions
  – calculation/code not yet in existence, but this is the end goal
- some trouble getting the simulations running
  – problems with tuning/processes called
  – software issues (discussed later)

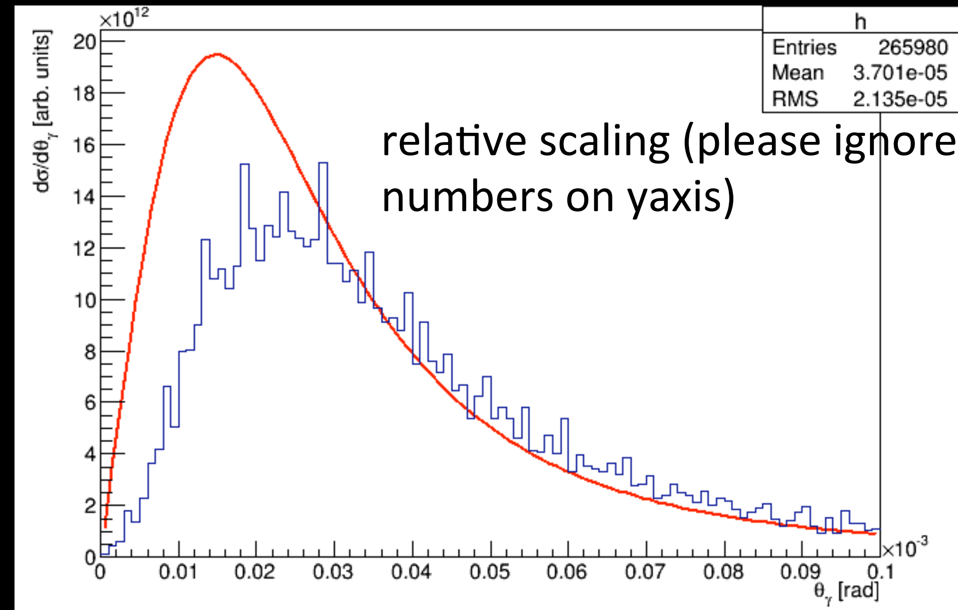
DJANGOH event simulation

- tuning parameters (some are modified from my input, I put the actual range used)
  - 1.3056e-11 < x < 0.99
  - 1e-10 < Q^2 < 1 GeV2^2
  - Wmin = 0.938 GeV
  - 1.3e-11 < y < 0.95

- caveats:
  - get the message: Warning: lower limit in x and/or Q2 too small for DIS formalism
  - lots of: Error in LYSTFU: x = +Infinity outside physical range
  - program enters SOPHIA module (for hadronization?) but fails (nothing to hadronize)
  - events seem to be mediated by Z_0 (pdg = 23) bosons, not photons!

Comparison of DJANGOH simulation with Bethe-Heitler





- observe some difference – still in the tuning?

** ELASTIC TAIL:

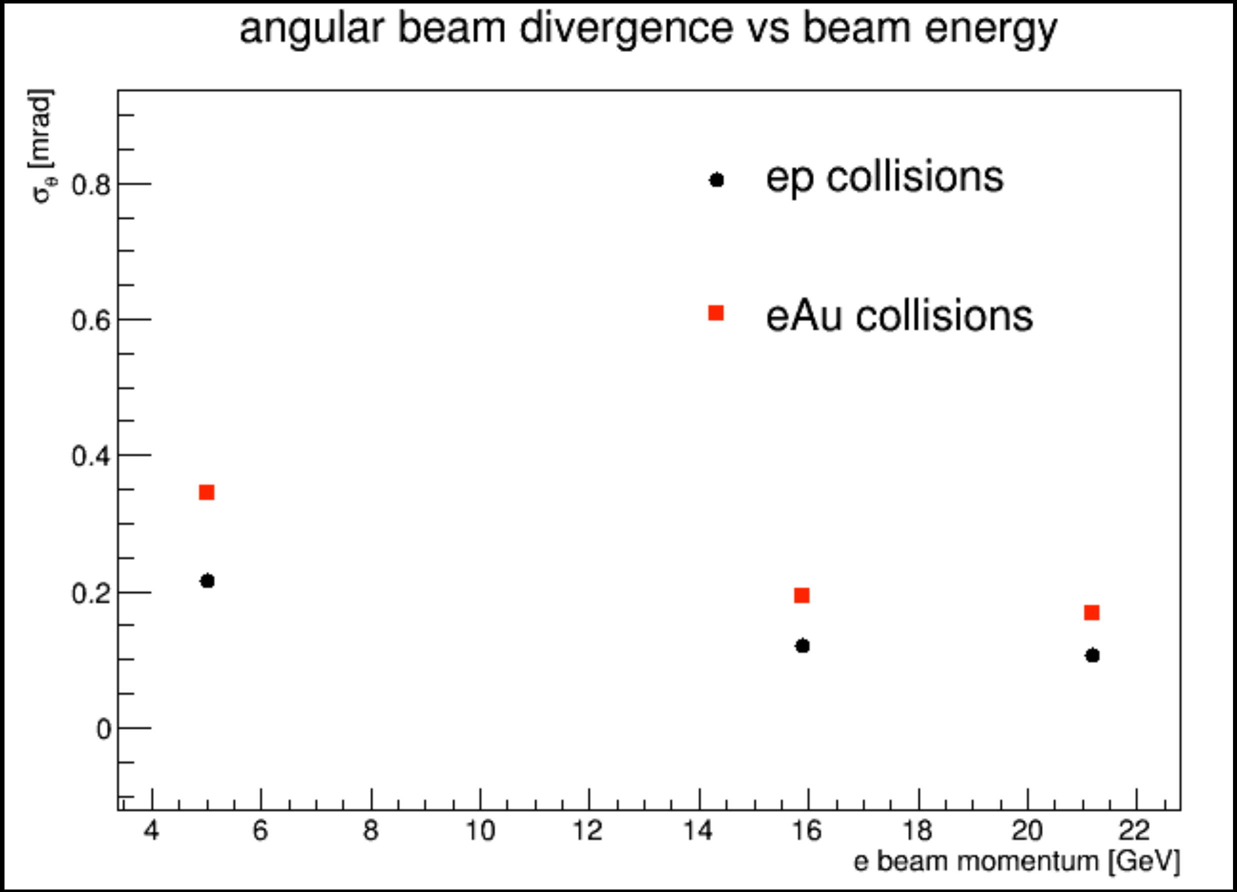
ELASTIC EP WITH CORRECTIONS	9.0308E+08 +/- 9.0308E+04 NB
INITIAL STATE LEPTONIC RADIATION	1.6597E+08 +/- 3.3521E+06 NB
FINAL STATE LEPTONIC RADIATION	1.6125E+08 +/- 1.1234E+06 NB
COMPTON CONTRIBUTION	3.9081E-01 +/- 3.5560E-01 NB

** TOTAL CROSS SECTION: 1.2303E+09 +/- 3.5365E+06 NB

Angular emittance of the beam

- taken from the eRHIC design study Table 3-1 (pg. 48)
  - emittance = 23e-6m (58e-6m) for ep (eAu)
  - beta* = 5cm

$$\sigma_\theta = \sqrt{\frac{\varepsilon}{\beta^* \gamma}}$$



Summary of needed luminometer acceptance

- from the machine:
  - 0.1 – 0.35 mrad from beam divergence
- from physical process:
  - 0.05 mrad (negligible)
- total
  - 0.4 mrad
  - still effects from crossing angle? small, see later slides

Code Developments

DJANGO debugging (I)

- the package as existing in the PACKAGES directory was not functioning
- first a complaint about a mismatch in version of lha library expected vs what is loaded
  - line 13476 of djangoh_h.f needed to change lhaverdef='5.8.6' to lhaverdef='5.9.1'
- segmentation fault due to line 2179 of lhaglue-5.9.1-copy.f
  - loop over nsets, but nsets is total number of pdf sets, which is large
  - causes loop to exceed the array length of mxset = 3
  - we initialize nset=0 and also change the loop to end at nmxset rather than nsets
  - this fixes the seg fault
- complaint of not finding polarized pdf (when running in unpolarized mode)
  - needed to modify function HSFG in django_u.f
  - change code so that function call to HSDPVR is only made if HPOLAR.ne.0, i.e. this function is only called if we are doing polarized beam collisions
- one more fix, found when compiling with –bounds-check
  - line 261 of djangoh_u.f
  - loop range was 0,10
  - Fortran arrays start at 1, not 0

DJANGO debugging (II)

- event record issue, no scattered proton in the record
  - seems that the scattered proton (for elastic events) is not written out in the event record
  - found a fix, modified sophia.f file near line 183
  - sophia gets called, but hadronization fails for this type of process (as it should for these elastic events)
  - added a statement that if there is nothing to hadronize, then just grab scattered proton information from the HEPEVT common block
- when going to much lower Q^2, received an error that stopped the program
  - received complaint that Q2min is outside the DIS regime
  - the LST(3) flag sets what to do in case of this type of warning
  - default is set to 2, which causes program to complain and stop execution
  - added a code word usable from the input file to change this (WARNING is the code word)
  - new value set LST(3) = 1, which causes the program to complain, but to continue running anyway

Code developments: eicsmear (I)

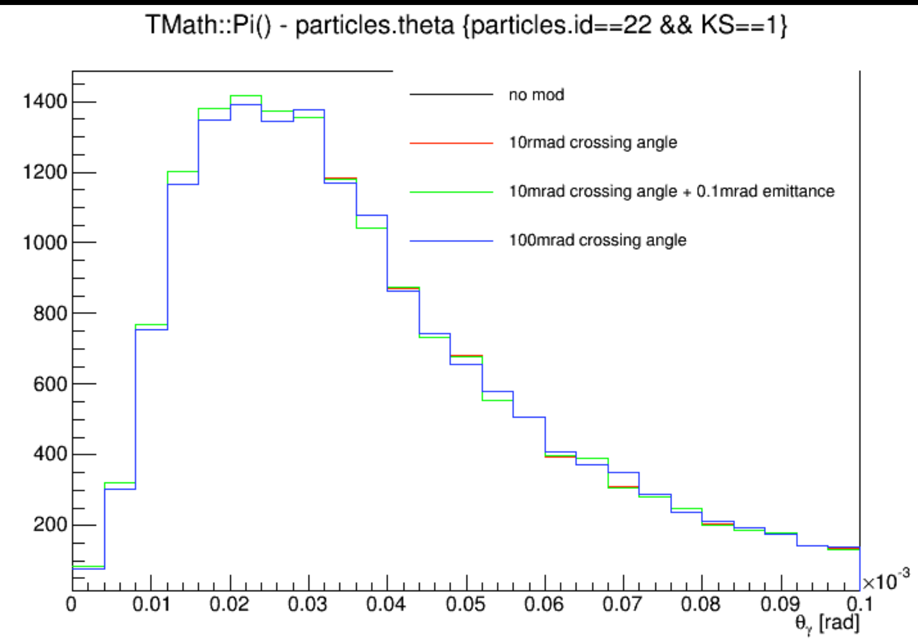
- further development of eicsmear (not in svn yet)
- new arguments can be passed to BuildTree() function
  - this function takes an input ASCII file from MC generators and produces a root file with a standard EICTree format
    - 1) argument to allow user to give each event a non-zero event vertex
    - 2) argument to allow user to give a crossing angle
    - 3) argument to allow user to give angular beam emittance
- a bit on the implementation:
  - added a new singleton class, EventVertexParameters
    - this class stores information related to the event vertex
  - an instance of this class can be called from anywhere in the program and always returns the same instance

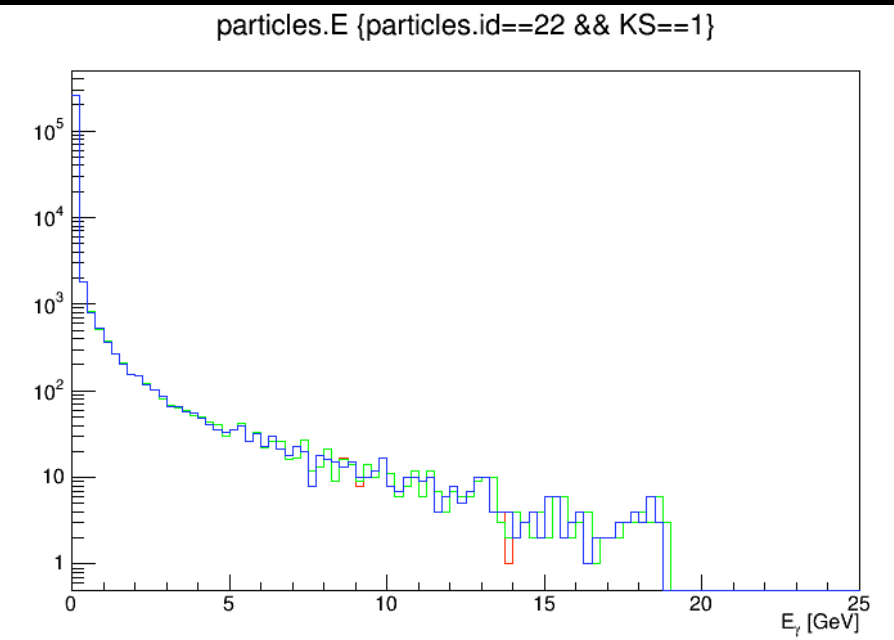
Code developments: eicsmear (II)

- event vertex smearing
  - input desired smearing parameters to arguments of BuildTree()
  - smears the x, y, and z position from user specified width
  - smear according to gaussian or uniform depending on users preference
  - this shifts the vertex position of ALL tracks in the event
  - stored in the event tree as xVertex, yVertex, zVertex
- crossing angle
  - ported Alexander's code into the eicsmear package
  - it calculates the boosts and rotations needed to transform the lab frame kinematics with a crossing angle to the lab frame scenario without a crossing angle
  - then the reverse operation of these boosts and rotations occur on the product particles of the collision
- angular beam divergence
  - the angular beam divergence is randomly generated on a per event basis based on the emittance provided by the user in the BuildTree() function
  - this generated angle is added to the crossing angle (which is negligible in comparison) before the boost and rotation matrices are calculated

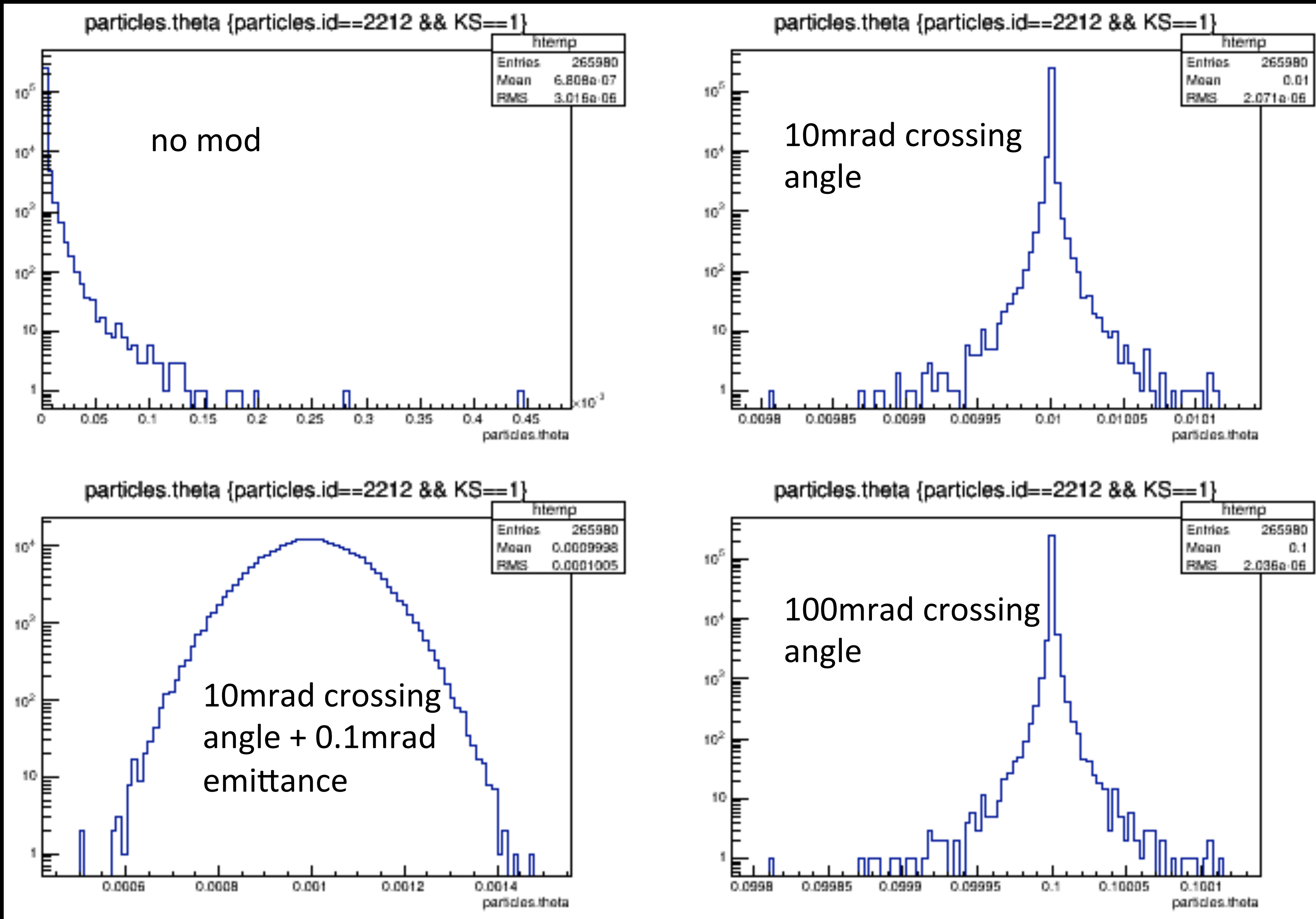
Effect of crossing angle?

- crossing angle leads to a negligible effect on angular distribution of photons
- note: the electron beam is taken along z-axis

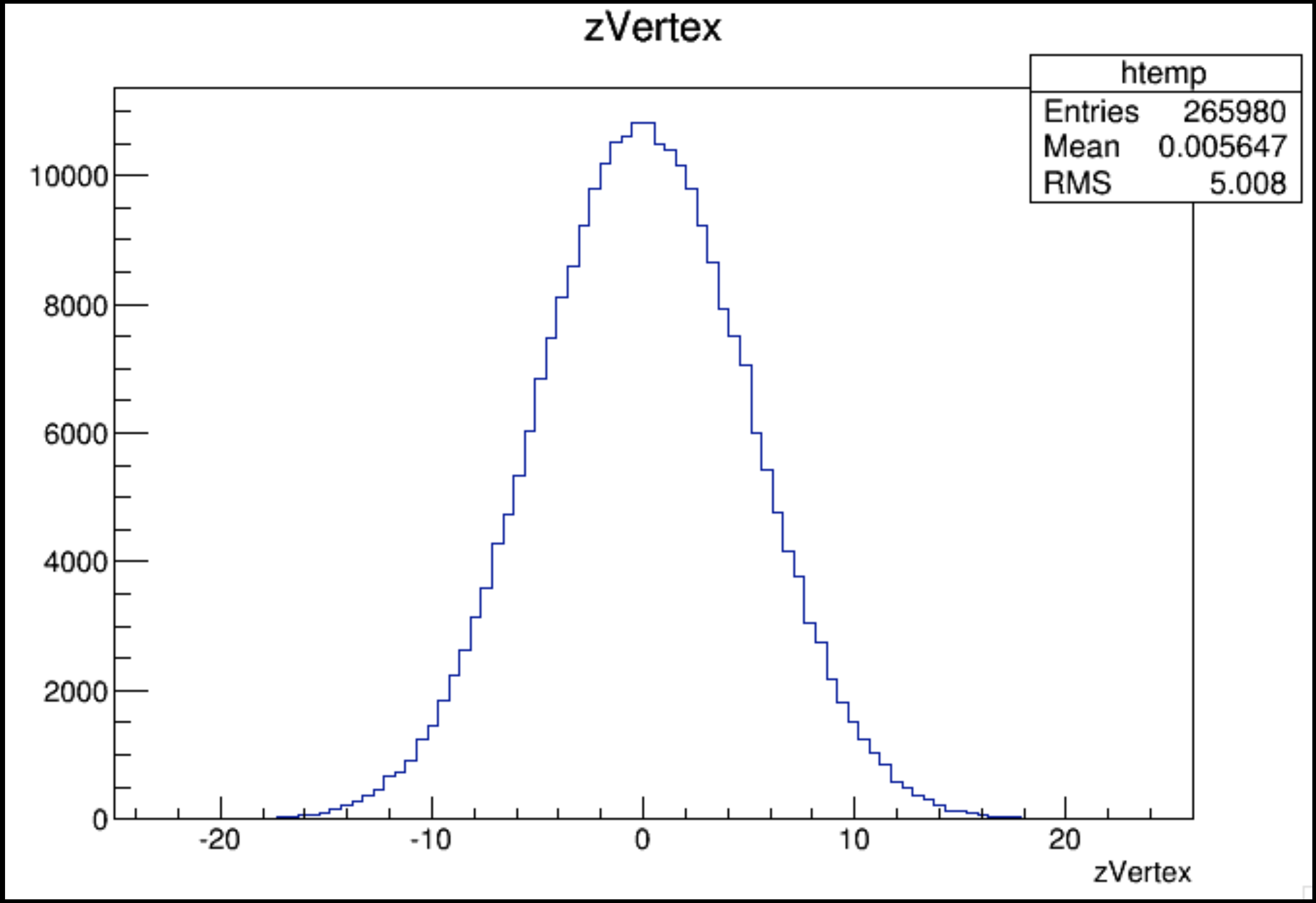




crossing angle effect on protons?



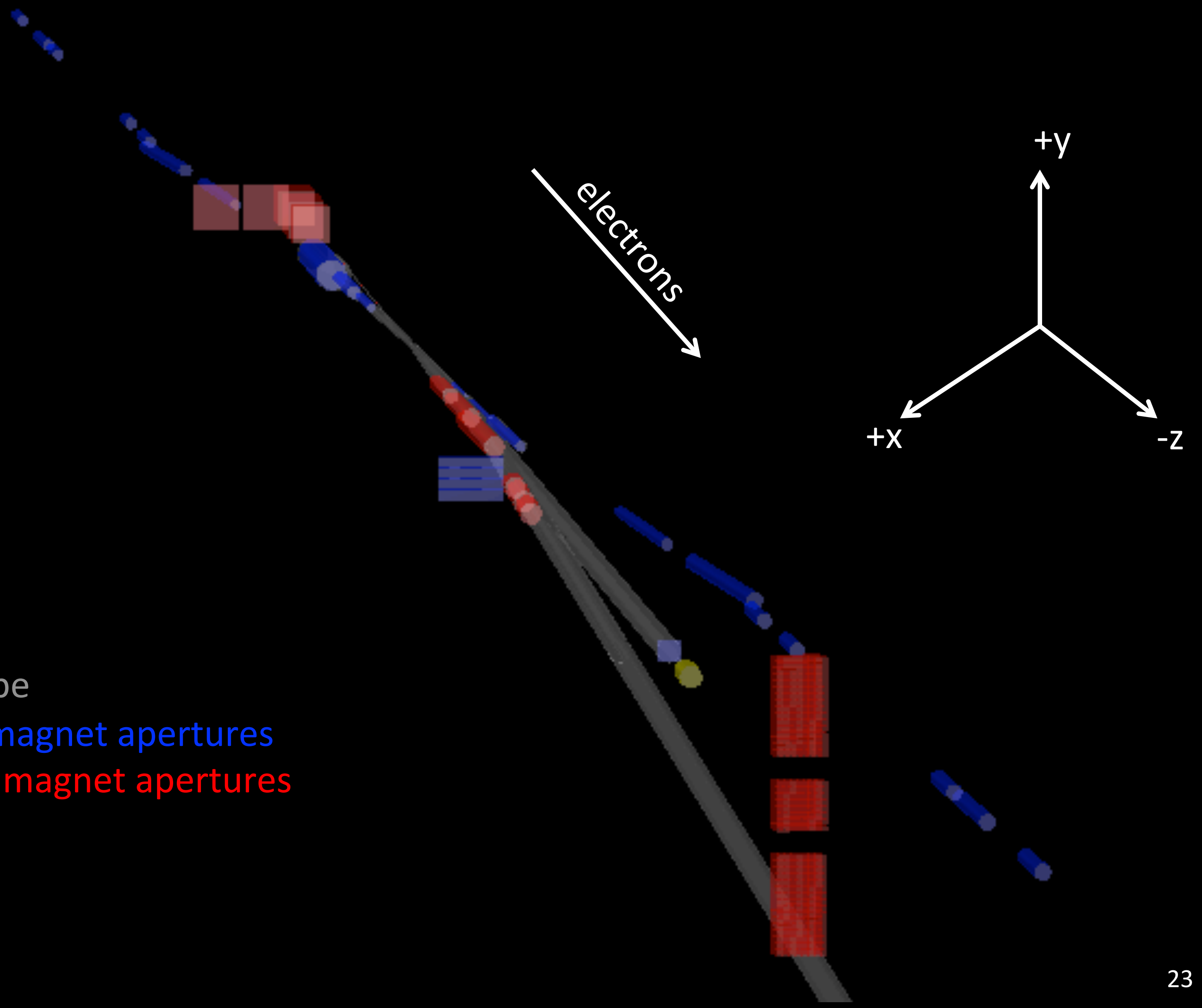
Proof of event vertex smearing



Code development: EicRoot

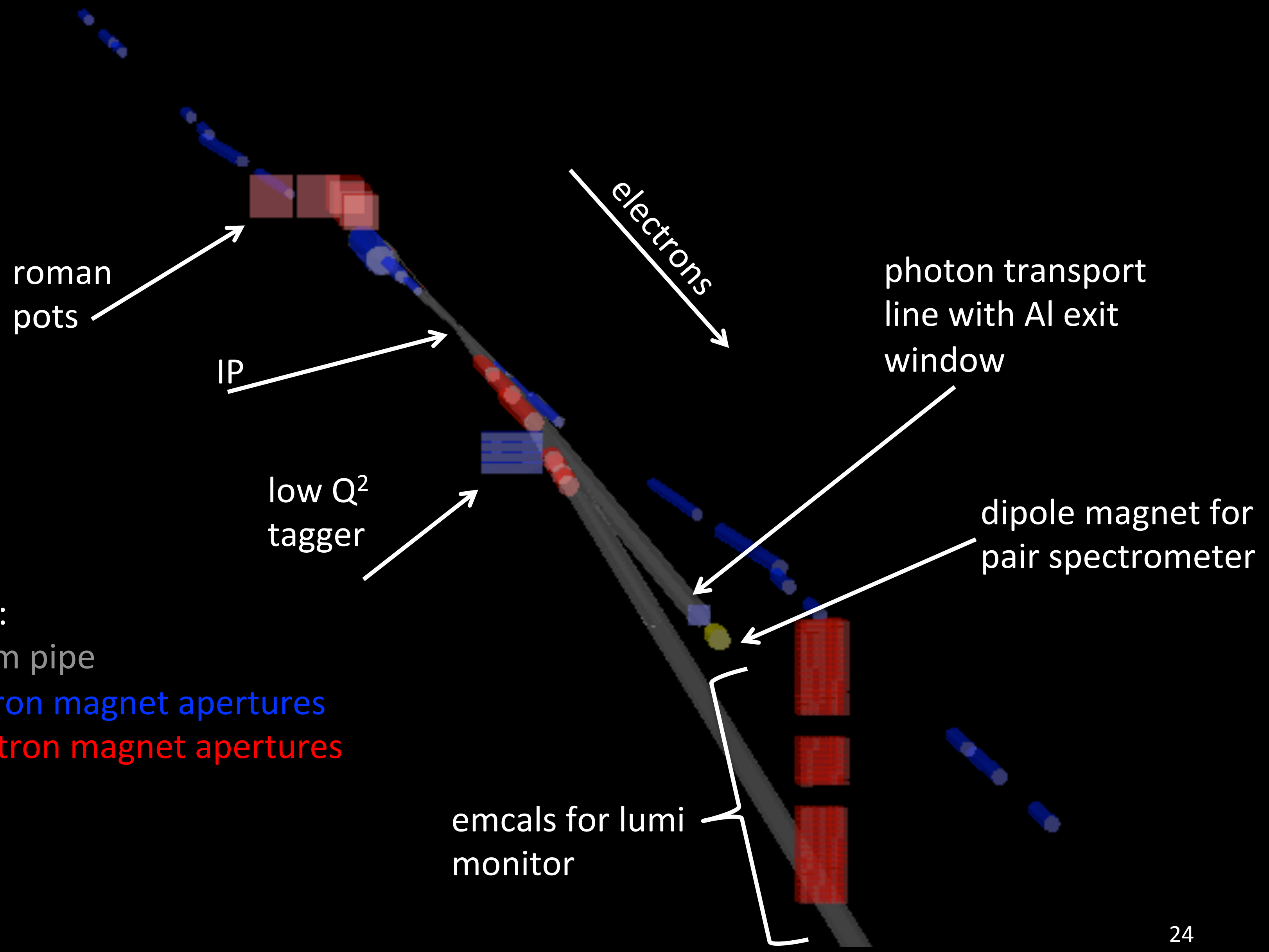
- Note: changes not yet pushed to SVN
- modified the functionality of the BeamPipeGeoParData class to allow for intersecting sections of beam pipe
- maybe the installation process can be streamlined for the user, but it works for now

General view of the current IR design implementation



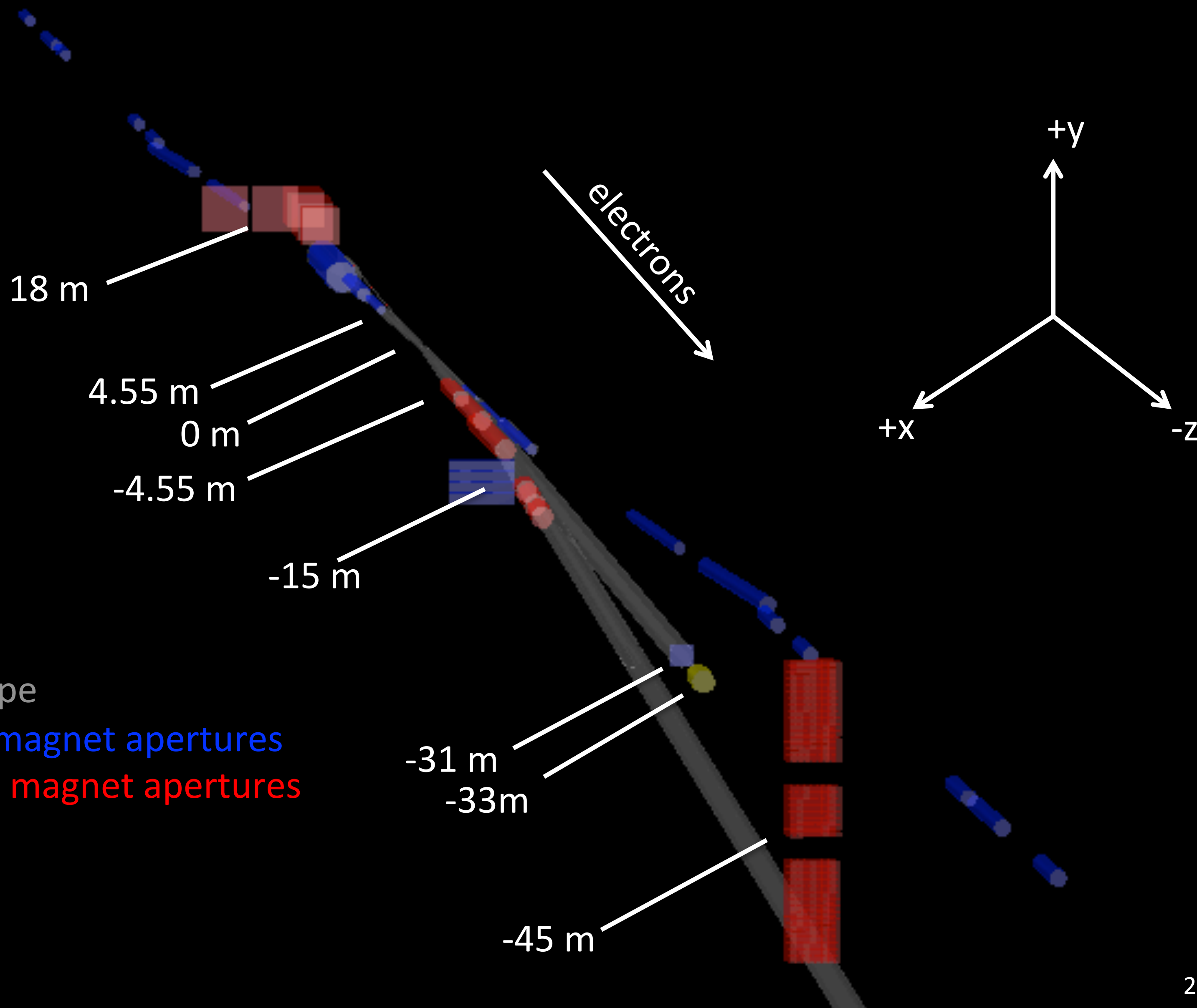
Legend:
- beam pipe
- hadron magnet apertures
- electron magnet apertures

General view of the current IR design implementation



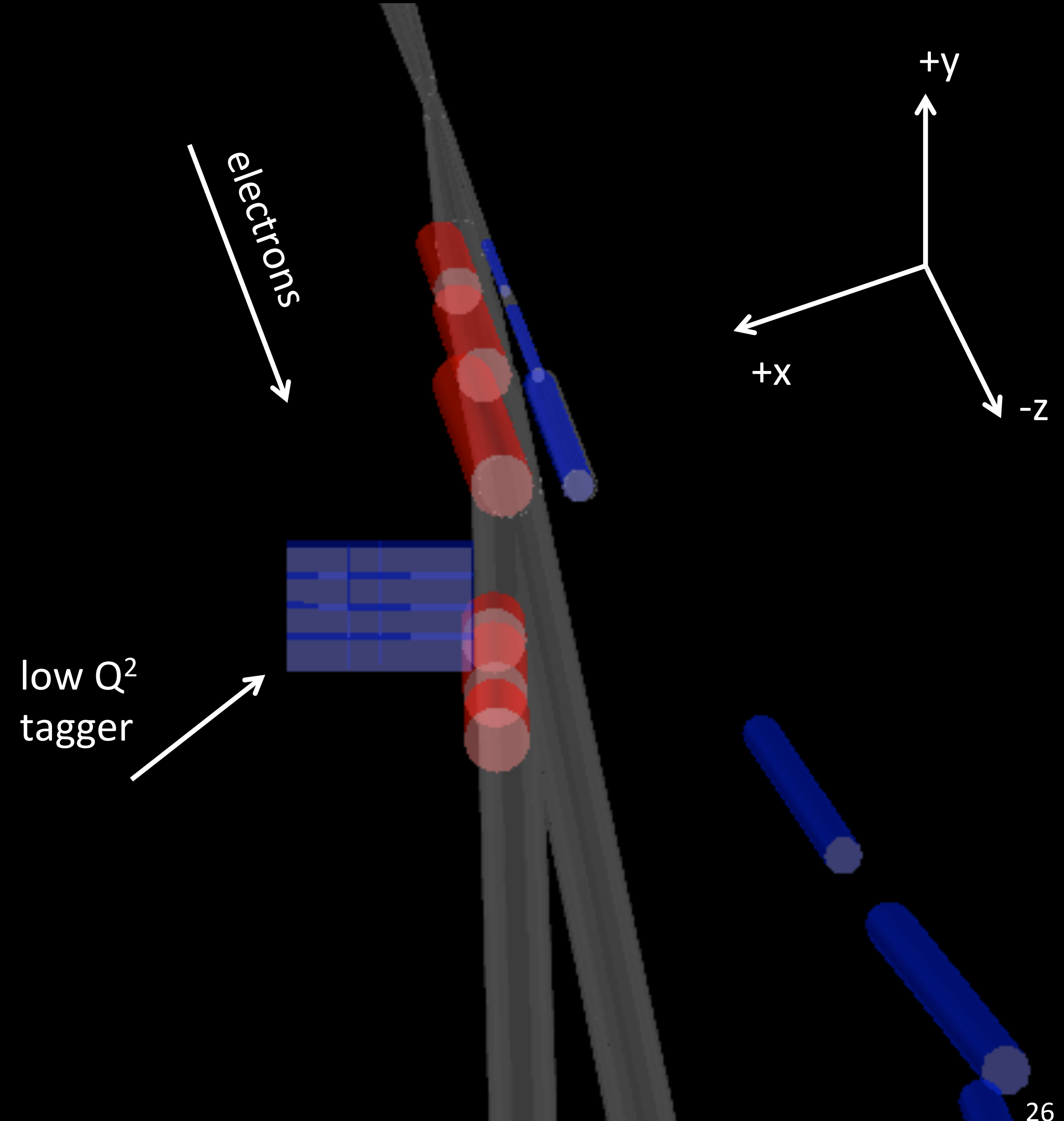
Legend:
- beam pipe
- hadron magnet apertures
- electron magnet apertures

General view of the current IR design implementation

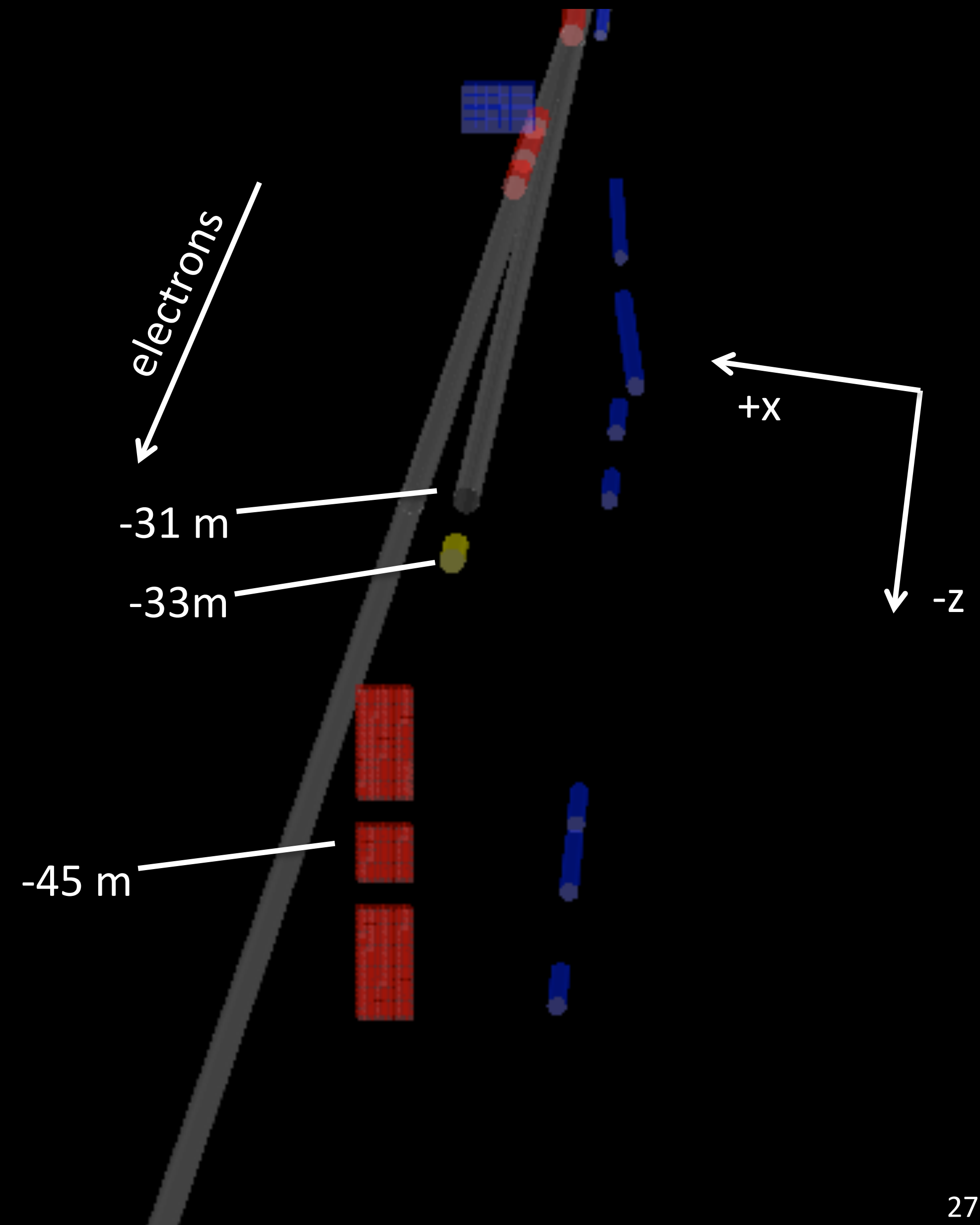


Legend:
- beam pipe
- hadron magnet apertures
- electron magnet apertures

Backward Direction (electron going side after the IP)



Backward Direction (electron going side after the IP)



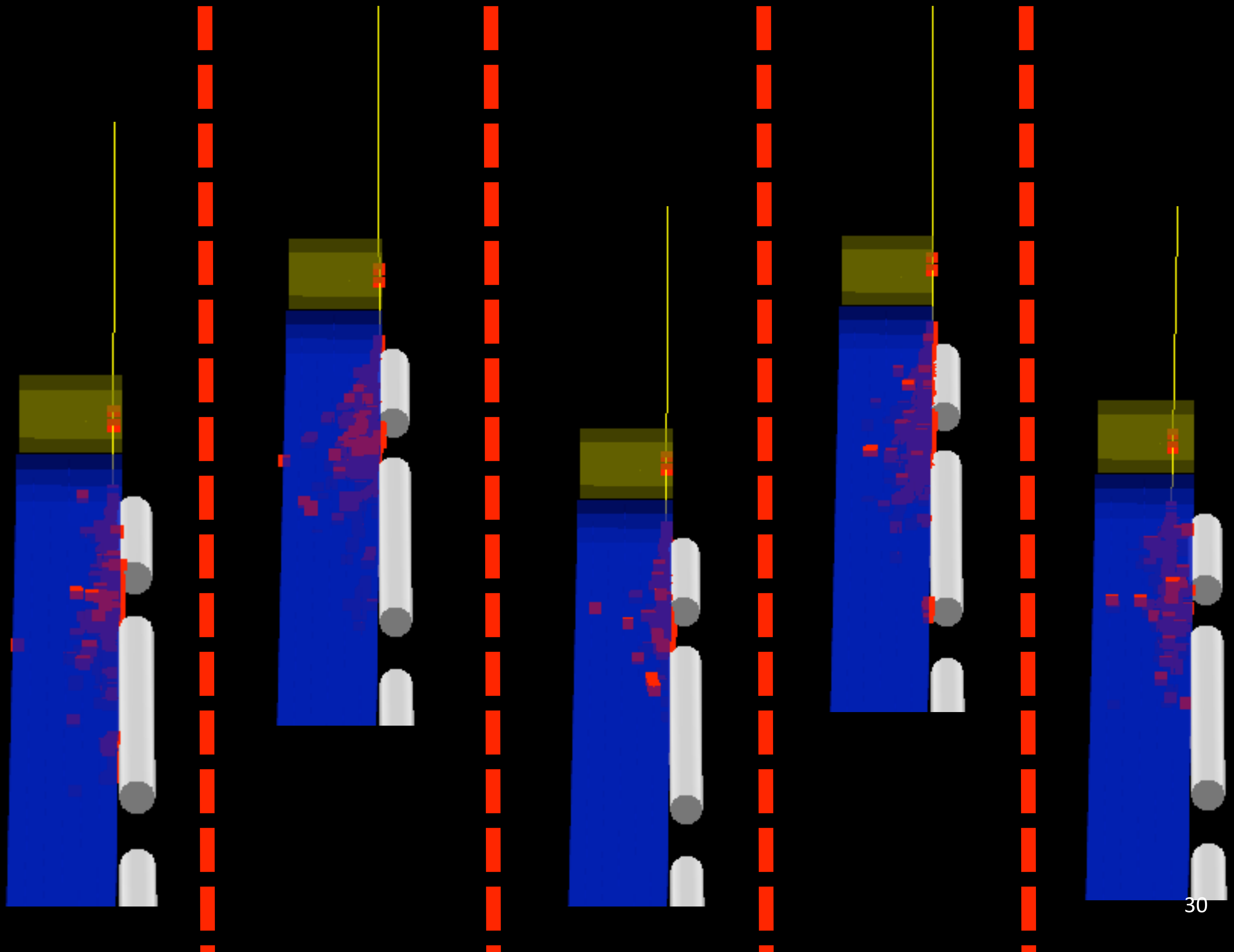
A little bit on the low Q^2 tagger…

- finally returning to the low Q^2 tagger simulation after working on the lumi design
- basic design implemented
- next step is to make the simulation more realistic
  - add a third tracking layer and do real track reconstruction
  - estimate the expected energy resolution given a realistic emcal design

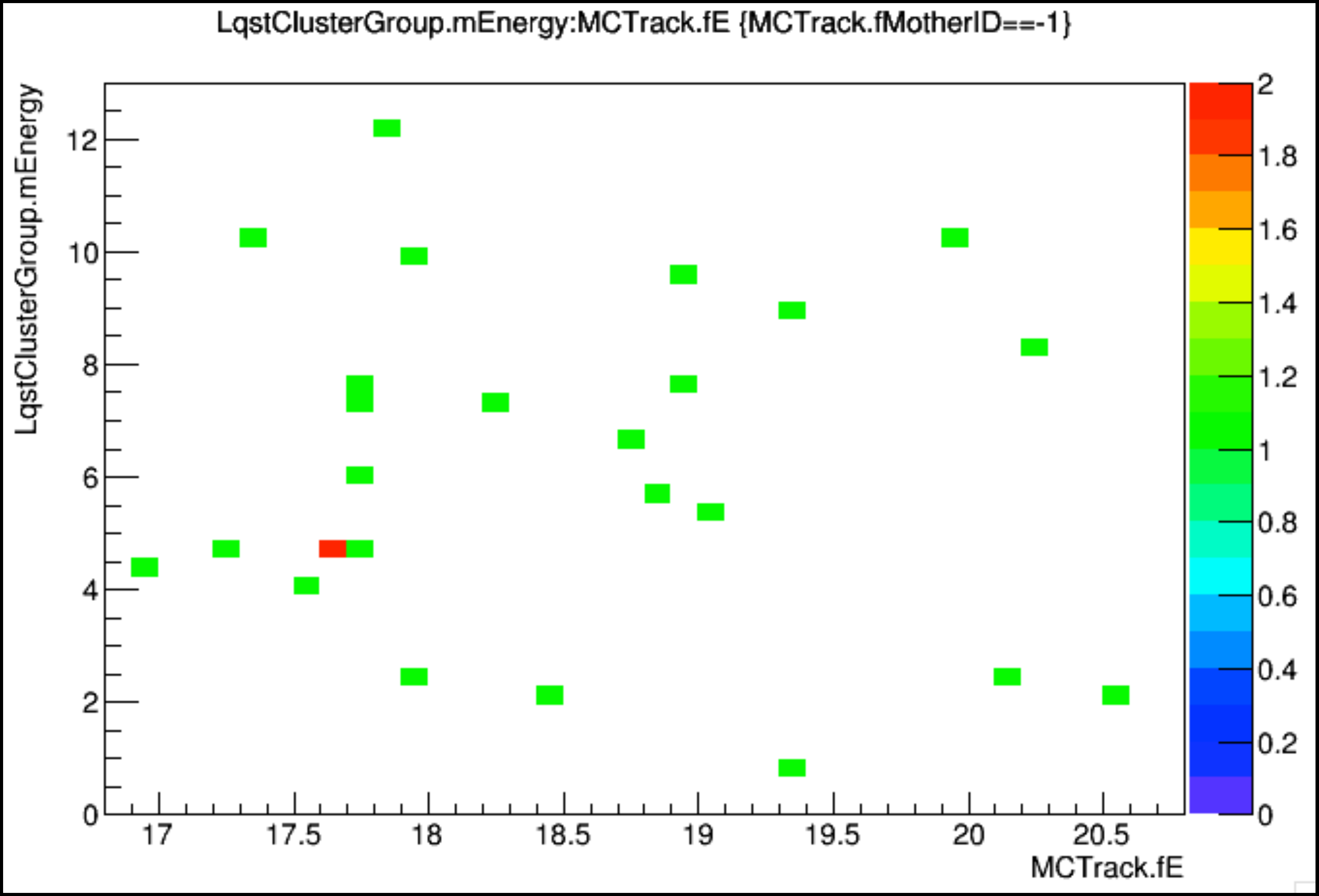
next iteration of the low Q^2 tagger design

- previous implementation of emcal was not very realistic
- was looking at pure MC hits in the previous iteration
  - Alexander did some code work on EicRoot to allow me to digitize hits for my combined tracker/emcal detector so that I can take the next step in sims
- implementation probably not long enough for full shower development
  - can play with materials and simulation parameters to adjust this, but this is probably true
- only considered active material before
  - now implemented a design with a crystal encased in inactive material more similar to a realistic design
  - play with the thickness of the absorber material
- increasing the tower length and looking to see the effect
- finding that lots of tracks actually deposit energy at the edge of the detector, causing loss of energy
  - screen shots on next slide
- will study the energy reconstructed as a function of distance from detector edge and electron energy
- will study the existing clustering algorithm in more detail
  - just running as a black box at the moment

some event displays



how well is the energy of the electron reconstructed?

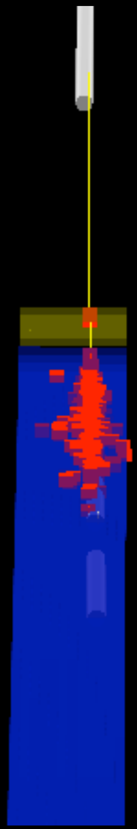


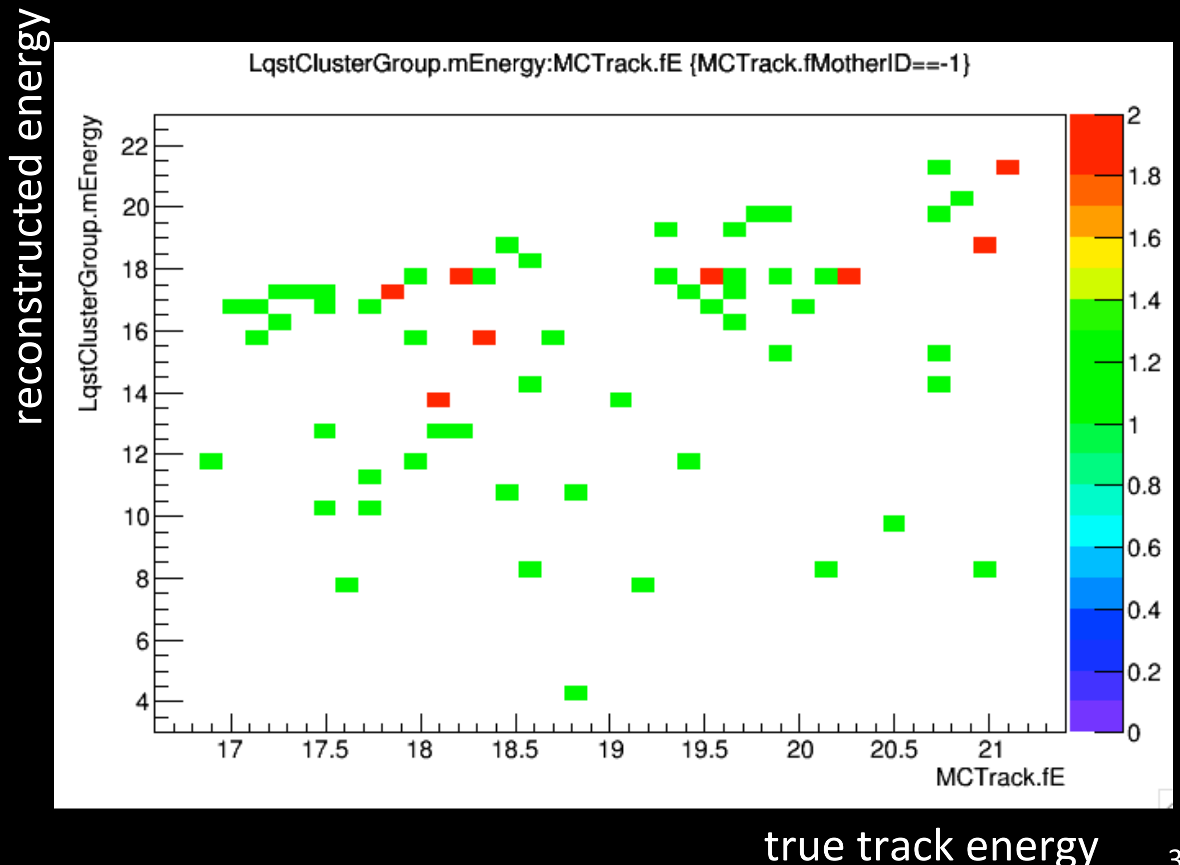
reconstructed energy

true track energy

test hypothesis of poor energy resolution because hits at edge

- simply widen and move detector over and see if things change
  - of course this cannot be realistically done since there is a beam there!!
- probably obvious, but it significantly helps the energy reconstruction





Transverse size of shower in emcal

- quantified by the Moliere radius

$$R_M = 0.0265 X_0 (Z + 1.2)$$

- need to contain shower to a few cm radius

Typical values for X_0, E_c and R_M of materials used in calorimeter

	X_0 [cm]	E_c [MeV]	R_M [cm]
Pb	0.56	7.2	1.6
Scintillator (Sz)	34.7	80	9.1
Fe	1.76	21	1.8
Ar (liquid)	14	31	9.5
BGO	1.12	10.1	2.3
Sz/Pb	3.1	12.6	5.2
PB glass (SF5)	2.4	11.8	4.3

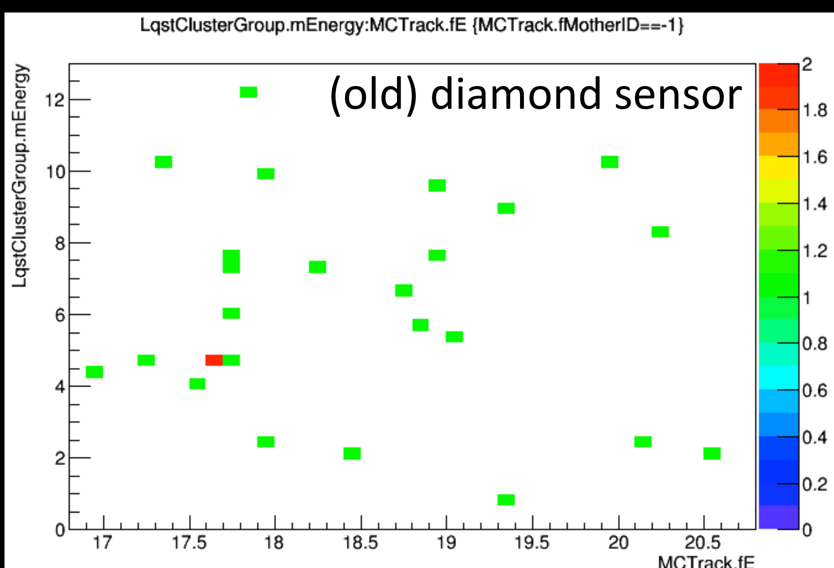
http://www.kip.uni-heidelberg.de/~coulon/Lectures/Detectors/Free_PDFs/Lecture9.pdf

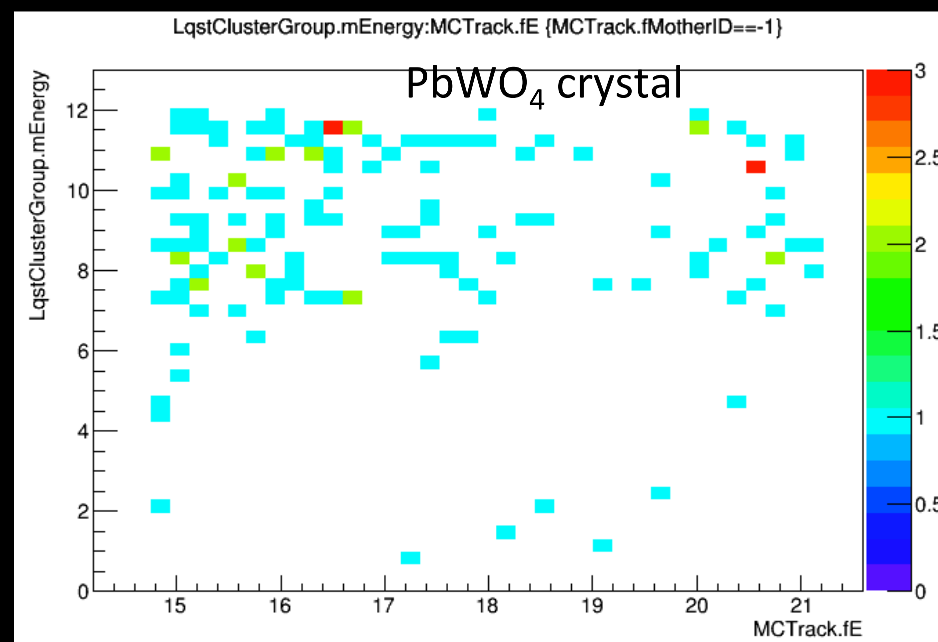
- CMS emcal:
  - lead tungstate ($PbWO_4$) cyrstals
  - R_M = 2.2cm
  - ρ = 8.3g/cm^3
  - 22x22 mm^2 at front
  - 26x26 mm^2 at back
  - 230 mm length
  - X_0 = 25.8
  - crystals contained within thin-walled glass-fiber alveola

http://cds.cern.ch/record/922757/files/lhcc-2006-001.pdf

Simulations with $PbWO_4$ ecal

- just implemented the $PbWO_4$ material this morning
- ran some quick simulations…definitely improved compared to previous (diamond sensor material) study, but still not great
- currently generating more events to look more differentially (function of scattering angle as well as energy)





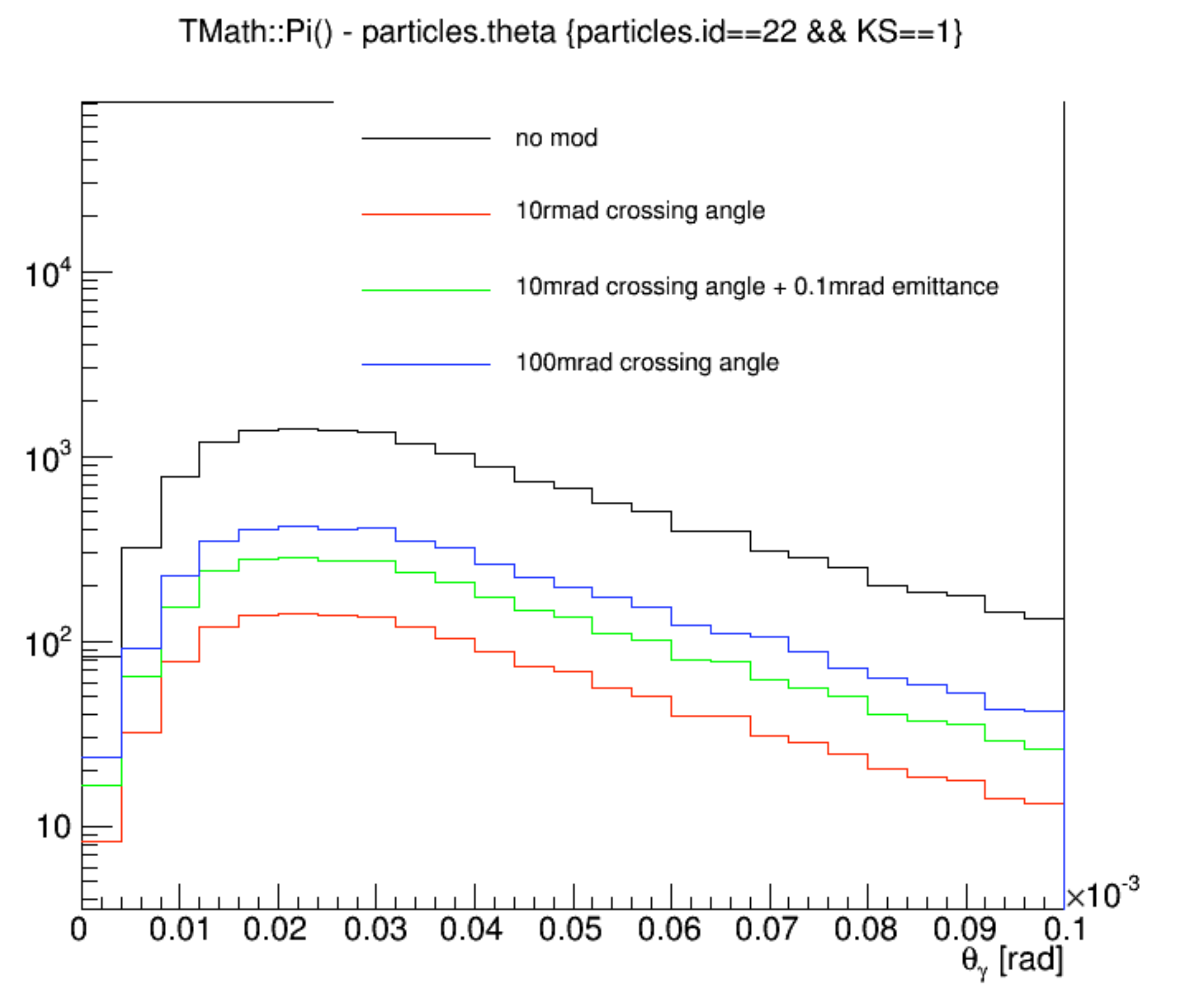
Summary

- Code developed for general simulation tools
  - DJANGOH
  - eicsmear
  - EicRoot
- Closing in on acceptance requirements for a luminosity monitor at eRHIC
- resuming detailed simulations of the low Q^2 tagger
- To do:
  - run some real event simulations to get acceptance for luminosity monitor
  - make beam pipe setup more automatic
    - define sections from magnet geometry files
  - return to low Q2 tagger simulations and get a more accurate measure of the resolution of the detector
    - carefully consider possible materials for the emcal ($PbWO_4$?)
  - roman pots – investigate effect of additional stations
  - think about a paper once things are at a sufficient level and some dust settles

Backups

Crossing angle details

- start with collision of ep, with e along z axis and p at an angle to the z axis
- boost the particles to the center of mass system, which has inherent angle from z axis
- rotate system so that particles align on z axis
- boost back to lab frame
- the matrices used to do this are stored in the EventVertexParameters class
- this process is reversed for the products of the collision


TMath::Pi() - particles.theta {particles.id==22 && KS==1}
no mod
10rmad crossing angle
10mrad crossing angle + 0.1mrad emittance
100mrad crossing angle
10
10^2
10^3
10^4
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
$\times 10^{-3}$
θ_γ [rad]